

Project IT Quant

Gaussian Process Regression for Derivative Portfolio Modélising and Application to CVA Computations

BA Ibrahima - EL JAMIY Mohamed - EL ADAMI Hicham

April 2021

1 Introduction

Les processus gaussiens (GP) sont une méthode générique d'apprentissage supervisé conçue pour résoudre des problèmes de régression. Le but de ce projet est d'étudier une approche de régression de processus gaussienne qui est bien adapté à la valorisation de portefeuille de dérivées OTC impliqué dans le calcul de CVA (Ajustement de valeurs de credit) basé sur l'article de Stéphane Crépey et Matthew F.Dixon intitulé "Regression de processus gaussien pour la modélisation de portefeuille dérivé et Applications aux calculs de CVA".

Dans cet article soumis à notre réflexion ,les auteurs utilisent la régression de processus gaussien pour déterminer de façon efficace une autre approximation du CVA.

Dans les lignes qui suivent nous allons d'abords présenté en théorie le processus gaussien ensuite l'appliquer pour le calcul du CVA puis le comparer avec la méthode MC-reval.

2 Processus Gaussien Régression (GPR)

un Processus Gaussien (GP) est un modèle bayésien non paramétrique, il permet de généraliser la définition de la distribution gaussienne d'un espace de vecteurs de dimension fini à un espace de fonctions à dimension infini. GPR représente bien un problème d'apprentissage supervisé, à partir des observations $(X, Y) = \{(x_i, y_i) | i = 1, 2, \dots, n\}$, on cherche à estimer une fonction f telle que $Y = f(X)$ et dont la distribution à priori fait partie de cette classe de processus gaussiens. GPR nous permet de joindre nos prédictions avec des intervalles de confiance qui représentent un élément important de la prise de décisions.

2.1 Définition GP

On dit une fonction $f : \mathbb{R}^p \rightarrow \mathbb{R}$ est issue d'un processus gaussien dont μ est la fonction moyenne et k la fonction covariance (kernel), i.e. $f \sim \mathcal{GP}(\mu, k)$ si :

$$\forall x_1, x_2, \dots, x_n \in \mathbb{R}^p \text{ le vecteur } [f(x_1), f(x_2), \dots, f(x_n)] \sim \mathcal{N}(\mu, K_{X,X})$$

μ est un vecteur tel que $\mu_i = \mu(x_i)$. La matrice de covariance $K_{X,X}$ vérifie $(K_{X,X})_{i,j} = k(x_i, x_j)$ où k doit être symétrique semi-définie positive :

$$\forall x_k \in \mathbb{R}^p \quad \forall \xi_k \in \mathbb{R} \quad \sum_{i,j=1}^n k(x_i, x_j) \xi_i \xi_j \geq 0$$

Exemples classiques de fonction de covariance :

Squared Exponential kernel (SE) :

$$k(x, x') = \exp\left(-\frac{1}{2l^2} \|x - x'\|^2\right)$$

le paramètre l s'interprète l'éloignement de x à x' dans l'espace des variables explicatives pour que $f(x)$ et $f(x')$ deviennent non corrélées.

Ce noyau est infiniment différentiable, ce qui implique que les GP avec ce noyau comme fonction de covariance ont des dérivées quadratiques moyennes de tous les ordres, et sont donc très lisses.

Matern kernel (MA) :

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|x - x'\|}{l} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{\|x - x'\|}{l} \right)$$

Où ν et l deux paramètres non négatifs, Γ la fonction gamma, et K_ν la fonction de Bessel modifiée de seconde espèce. On peut montrer que ce kernel converge vers SE quand $\nu \rightarrow +\infty$

Les deux kernels font partie de Fonctions de Base Radiale (RBF) qui ne dépendent que de $\|x - x'\|$

2.2 Estimation

L'estimation des paramètres du kernel se fait par maximisation de la vraisemblance marginale :

$$\log p(Y|X, \lambda) = -[Y^T (K_{X,X} + \sigma^2 I)^{-1} Y + \log \det(K_{X,X} + \sigma^2 I)] - \frac{n}{2} \log(2\pi)$$

avec λ regroupe l'ensemble des hyperparamètres du kernel et inclut aussi σ ($\lambda = [l, \sigma]$ pour SE kernel). Dans la pratique, le gradient stochastique est utilisé dans la recherche de $\lambda^* = \arg \max_\lambda \log p(Y|X, \lambda)$:

$$\partial_\lambda \log p(Y|X, \lambda) = \text{tr}(\alpha \alpha^T - (K_{X,X} + \sigma^2 I)^{-1}) \quad \partial_\lambda (K_{X,X} + \sigma^2 I)^{-1} \quad \text{où} \quad \alpha := (K_{X,X} + \sigma^2 I)^{-1} Y$$

Pour SE kernel :

$$\begin{cases} \partial_\sigma (K_{X,X} + \sigma^2 I)^{-1} = -2\sigma (K_{X,X} + \sigma^2 I)^{-2} \\ \partial_l (K_{X,X} + \sigma^2 I)^{-1} = -(K_{X,X} + \sigma^2 I)^{-2} \quad \partial_l K_{X,X} \quad \text{avec} \quad \partial_l k(x, x') = l^{-3} \|x - x'\|^2 k(x, x') \end{cases}$$

2.3 Prédiction

On suppose un bruit gaussien iid, $y|x \sim \mathcal{N}(f(x), \sigma^2)$. Sachant les données d'apprentissage (X, Y) , la distribution prédictive du GP évaluée sur des nouvelles observations X_* s'écrit :

$$f_*|X, Y, X_* \sim \mathcal{N}(E[f_*|X, Y, X_*], \text{Var}[f_*|X, Y, X_*])$$

Où :

$$\begin{cases} E[f_*|X, Y, X_*] = \mu_{X_*} + K_{X_*,X} [K_{X,X} + \sigma^2 I]^{-1} Y \\ Var[f_*|X, Y, X_*] = K_{X_*,X_*} - K_{X_*,X} [K_{X,X} + \sigma^2 I]^{-1} K_{X,X_*} \end{cases}$$

$K_{X,X}, K_{X_*,X}, K_{X,X_*}$ et K_{X_*,X_*} sont obtenu en appliquant la fonction de covariance $k : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ aux observations qui correspondent X ou X_*

2.4 Complexité

Le temps d'apprentissage du modèle augmente avec le nombre d'observations, la maximisation de la vraisemblance marginale revient à la résolution d'un système linéaire et au calcul du déterminant d'une matrice $n \times n$ ce qui introduit une complexité $\mathcal{O}(n^3)$, Or la phase prédiction se fait à $\mathcal{O}(n^2)$. **Massively Scalable Gaussian Processes (MSGP)** est une extension qui a permet de réduire significativement la complexité et accélérer l'apprentissage et la prédiction du GP.

2.5 Exemple d'utilisation de GP sur BS

on veut determiner le prix du call dans le modèle de Black Scholes(BS) en utilisant GP implémenté dans scikit learn. Pour cela on définit une fonction lambda : $(\text{lambda } x : \text{bsformula}(1, \text{lb}+(\text{ub}-\text{lb})x, \text{KC}, r, T, \text{sigma}, 0)[0])$ qui calcule analytiquement le prix du call. où lb,ub est respectivement la borne inférieur et supérieur sur le domaine, r :le taux d'intérêt sans risque, K :prix d'exercice, sigma : la volatilité.

Les données d'entrainement(training set) sont : *train_x_1* et *train_y_1*.

train_x_1 contient 5=*training_samples* (d'echantillon d'apprentissage) nombres compris entre 0 et 1 générés grâce à np.linspace.

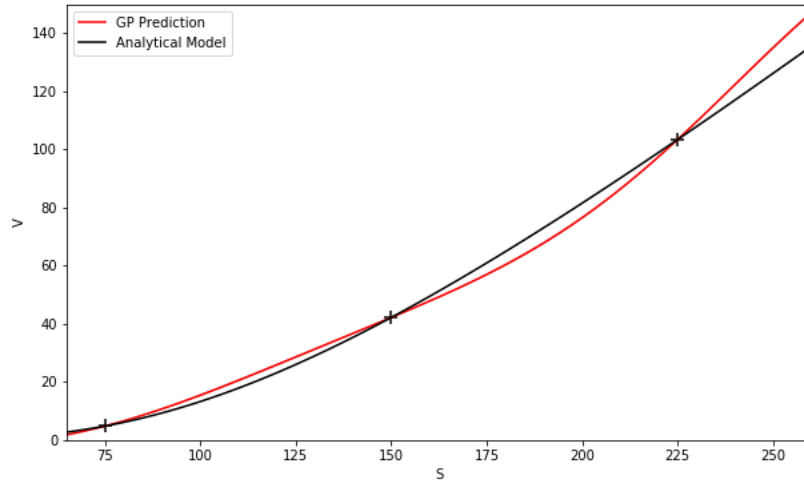
train_y_1 contient les prix du call calculés avec la fonction bsformula définit ci dessus qui sont au nombre de 5.

de même on définit les données de test (test set) qui sont : *test_x_1* et *test_y_1* générés de la même manière de taille (100,100).

après avoir définit correctement le training set et le test set on peut appliquer le modèle GP (Gaussian-ProcessRegressor) implémenté dans scikit learn avec un noyau RBF aux training set.

une fois l'entrainement terminé on va prédire de nouvelle données *y_pred* en utilisant la méthode predict qui prend comme paramètre *test_x_1*.

En traçant les deux courbes on constate qu'on a une bonne approximation juste en utilisant que 5 données d'entrainements.



On voit bien l'efficacité de GP sur le calcul du prix du call dans le modèle de Black Scholes. On pourrait améliorer l'approximation en augmentant le nombre de training set car plus on a suffisamment de données d'entraînement plus on augmente l'efficacité du modèle.

3 Application du modèle GP sur le CVA

Le CVA est défini, au niveau d'un portefeuille ou d'un contrat, comme étant la différence entre la valorisation sans risque et la valorisation qui tient en compte la probabilité de défaut. Donc calculer la CVA sous-entend une modélisation du risque de défaut préalable.

Le CVA permet de déterminer la "fair value" de certains produits dérivés et de mettre en place des réserves pour se prémunir du défaut d'une contrepartie. En pratique elle est calculée indépendamment pour chaque contrepartie puis agrégée pour obtenir une CVA globale par portefeuille. On a vu dans le cours d'IT Quant qu'on pouvait numériquement calculer le CVA_0 par simulation imbriquée en appliquant la méthode de Monte Carlo qui est coûteux en terme de calcul (généralement programmer en GPU).

On veut étudier une autre méthode de calcul qui utilise la régression de processus gaussien pour déterminer une approximation efficace du CVA.

3.1 Calcul du CVA

Dans cette section, à titre d'exemple d'application du risque de portefeuille, nous considérons l'estimation de risque de crédit de contrepartie sur un portefeuille client. La perte attendue pour la banque associée à la contrepartie en défaut est donnée par le CVA.

Differentes formules pour le calcul de ce dernier sont données par :

$$CVA_0 = (1 - R) \mathbf{E} \int_0^T \beta_t \pi_t^+ \delta_\tau (dt)$$

où δ_t est la mesure de Dirac du client au temps de défaut τ , R est le taux de récupération (recovery rate) et β_t étant le processus de remise.

Admettons que le temps d'arrêt τ soit doté d'un processus d'intensité stochastique γ et d'une configuration d'immersion de base entre la filtration du marché et la filtration progressivement agrandi avec τ , on a donc une deuxième formule :

$$CVA_0 = (1 - R)\mathbf{E} \int_0^T \beta_t \pi_t^+ e^{-\int_0^t \gamma_s ds} \gamma_t dt$$

Selon les spécifications markoviennes, π_t est une fonction déterministe du temps t et des facteurs de risques X_t c'est à dire $\pi(t, X_t)$ et de même dans le cas des modèles d'intensité, $\gamma_t = \gamma(t, X_t)$. Les facteurs communs à π et γ permettent de modéliser le risque dans le mauvais sens c'est à dire le risque de dépendance indésirable entre le risque de défaut du client et l'exposition de marché correspondante.

Dans le cas particulier où le défaut est indépendant de la valeur du portefeuille exprimée en unités numériques, alors la première formule ci-dessous peut être exprimé par :

$$CVA_0 = (1 - R) \int_0^T \mathbf{E}[\beta_t \pi_t^+ p(t)] dt$$

où $p(t)$ est la fonction de densité de probabilité de τ . Pour calculer le CVA numériquement basé dans l'équation précédente c'est à dire le cas indépendant, on discrétise l'intervalle en temps en N dates $t_1 = 0, t_2, \dots, t_N = T$ à travers laquelle on évalue EPE(Expected Positive Exposure), $\mathbf{E}[\beta_t \pi_t^+]$. Les probabilités $\Delta p_i = P(t_i < \tau < t_i + 1)$ peuvent être amorcés à partir de la courbe CDS du client.

Nous allons d'écrire la méthode de MC- Gaussian Processus (GP) utilisé dans cet article.

3.2 Description de la méthode MC-GP

Le but est d'utiliser maintenant une approximation rapide qui utilise la regression de processus gaussien pour la valorisation de π . Contrairement à ce qu'on a vu pendant le cours d'IT Quant où on utilise la méthode de simulation imbriquée utilisant Monte Carlo.

Dans ce document on utilise ce qu'on appelle la méthode MC-GP (Monte Carlo -Gaussian Processus) pour le calcul du CVA c'est à dire un métamodèle pour π est ajusté pour modéliser les données générées, en admettant une génération de processus de données des facteurs de risques X_t . Par la suite l'apprentissage automatique est utilisé pour apprendre les expositions dérivées des composantes en tant que fonction des paramètres sous-jacents, y compris le délai de maturité T et les calculs CVA qui en résultent sont ensuite effectués par simulation de Monte Carlo basée sur ce métamodèle pour π .

Cette procédure est alors appelée approche de calcul MC-GP CVA. La formule de l'estimation du MC-GP dans le cas indépendant décrit dans le document est donnée par :

$$CVA_0 = \frac{(1 - R)\Delta t}{M} \sum_{j=1}^M \sum_{i=1}^N \pi(t_i, X_{t_i}^j) \beta_{t_i}^j \Delta p_i$$

où l'intervalle en temps et en espace sont discrétisés de taille respectivement N et M , Δt le pas de temps, la valeur exacte du portefeuille $\pi(t_i, X_{t_i}^j)$ est évalué pour le facteur de risque simulé $X_{t_i}^j$ évalué à l'espace j et au temps t_i .

Puisqu'on cherche une estimation par un processus de regression gaussien on remplace la valeur exacte du portefeuille défini ci-dessus par la moyenne de la fonction postérieure conditionnée par la simulation du

risque de facteurs du marché X_{t_i} , donc en remplaçant π par l'esperance de la moyenne conditionnelle décrit précédemment l'équation devient :

$$C\hat{V}A_0 = \frac{(1-R)\Delta t}{M} \sum_{j=1}^M \sum_{i=1}^N \beta_{t_i}^j (\mathbf{E}[\pi_* | X, Y, x_* = X_{t_i}^j])^+ \Delta p_i$$

dans le cas le cas de l'intensité stochastique, la formule ci-dessous devient :

$$C\hat{V}A_0 = \frac{(1-R)\Delta t}{M} \sum_{j=1}^M \sum_{i=1}^M \beta_{t_i}^j (\mathbf{E}[\pi_* | X, Y, x_* = X_{t_i}^j])^+ e^{-\Delta t \sum_{l < i} \gamma(t_l, X_{t_l}^j)} \gamma(t_l, X_{t_l}^j)$$

L'erreur d'échantillonnage MC dans l'estimation GP-MC est donnée par :

$$error = \frac{1}{M-1} \sum_{j=1}^M [(1-R)\Delta t \sum_{i=1}^N \beta_{t_i}^j (\mathbf{E}[\pi_* | X, Y, x_* = X_{t_i}^j])^+ e^{-\Delta t \sum_{l < i} \gamma(t_l, X_{t_l}^j)} \gamma(t_l, X_{t_l}^j) - C\hat{V}A_0]$$

3.3 Comparaison entre la méthode MC-reval et GP-MC

Puisque la méthode de MC-GP se différencie de MC-reval(avec repricing en utilisant les formules de Black Scholes de portefeuille) au niveau de l'évaluation de PEE(Positive Expected Exposure) du portefeuille à travers le temps.

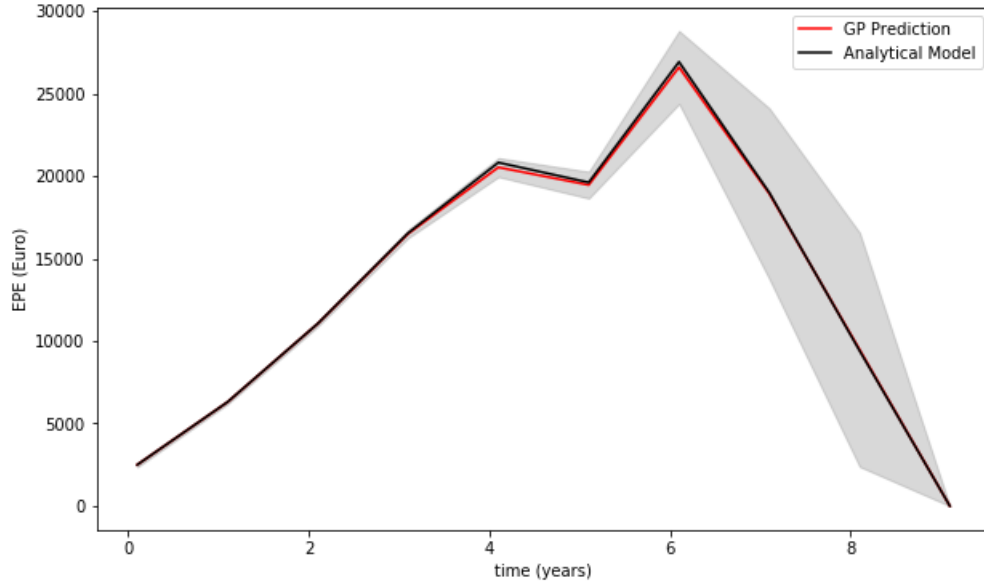
Pour MC-GP on determine la valeur du portofeuille π en faisant une estimation de la regression de processus gaussien alors que celui de MC-reval on utilise la valeur du portefeuille donné dans l'équation de Black Scholes avec comme paramètre : S, K , r ,time dans l'implémentation de la méthode bsformula.

On donne les valeurs pour M=1000, N=100 et $S_0=100$ pour la simulation du facteur de risque du marché.

En introduisant l'intensité dynamique suivante $\gamma(S_t) = \gamma_0(\frac{S_0}{S_t})^{\gamma_1}$, où $(\gamma_0, \gamma_1) = (0.02, 1.2)$ Le CVA est calculé sur la base de l'équation

$$C\hat{V}A_0 = \frac{(1-R)\Delta t}{M} \sum_{j=1}^M \sum_{i=1}^M \beta_{t_i}^j (\mathbf{E}[\pi_* | X, Y, x_* = X_{t_i}^j])^+ e^{-\Delta t \sum_{l < i} \gamma(t_l, X_{t_l}^j)} \gamma(t_l, X_{t_l}^j)$$

Le graphe suivant donne l'EPE (expected positive exposure) estimée du portefeuille (c'est à dire π_0) des deux méthodes (modèle Analytique versus GP prediction) en fonction du temps :



on voit que les deux graphes sont quasiment confondues. La bande grise désigne l'intervalle d'incertitude de 95% du GP. On peut dire on a quasiment une bonne approximation.

De même après avoir représenté PEE, cette fois on détermine la valeur CVA_0 estimé en fonction du nombre de scénarios.

Pour cela une méthode `CVA-simulation(sim-params, model-params, def-model)` a été défini dont le but est de calculer la valeur du CVA_0 et de π_0 . On va détailler ce que prennent les paramètres de la méthode définie ci-dessus.

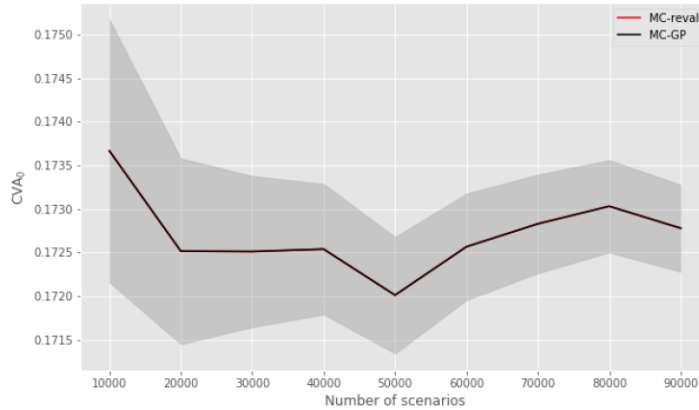
Le paramètre `sim-params` est un dictionnaire dont les éléments sont : number of Euler stpes, number of paths, number of exposure dates et time grid of exposure dates.

Le paramètre `model-params` est un dictionnaire dont les éléments sont : risk-free rate (r), volatility (σ), maturity (T), S_0

Le paramètre `def-model` (paramètre pour l'intensité de défaut du modèle) est un dictionnaire dont les éléments sont : $\gamma_0 = 0.01$, $\gamma_1 = 1.2$ et R (recovery).

Après avoir défini les paramètre, le modèle `GaussianProcessRegressor` est importé à partir de Scikit Learn sous python avec comme noyau le RBF, également connu sous le nom de noyau «exponentiel au carré». Puis on l'entraîne sur les données x_{train} et y_{train} . Une fois que l'entraînement est fini on prédit des nouvelles valeurs que le modèle ne connaît pas.

la courbe ci-dessous nous donne le CVA_0 estimé dans les deux modèles en fonction du nombre de scénarios.



On voit que les deux courbes estimées de MC-reval et MC-GP sont quasiment confondues d'après le graphe. Et il montre les propriétés de convergence de Monte Carlo du CVA_0 estimé par méthode du MC-GP.

4 Conclusion

Cet article présente une approche du MC-GP (Monte Carlo-Gaussian Process) pour une évaluation rapide des portefeuilles dérivés, de leurs sensibilités et les mesures du risque entre les parties telles que l'EPE (expected positive exposure) et le CVA.

On rappelle que EPE a été évalué à partir de π_0 et CVA à partir de CVA_0 dans la méthode *CVA_computation*.

MC-GP est très efficace comparé aux réseaux de neurones profonds (DNN) car il requiert beaucoup moins de données pour l'entraînement du modèle.

L'article nous a permis de découvrir une nouvelle approche de calcul du CVA qui est très différente de ce qu'on a vu dans le cours IT Quant, car au niveau du cours on a vu l'approche par Simulations imbriquées dont on a utilisé la méthode de Monte Carlo doublement qui est très lourde en terme de calcul mais aussi celle des réseaux de neurone.

5 Bibliographie

Gaussian Process Regression for Derivative Portfolio Modeling and Application to CVA Computations
Stephane Crepey LaMME, Univ Evry, CNRS, Universite Paris-Saclay, 91037, Evry, France and Matthew
F.Dixony Department of Applied Mathematics, Illinois Institute of Technology. October 18, 2019