Kernel-Based Generative Models
with Applications to Risk Management in Finance

Jean-Marc Mercier

MPG-Partners *jean-marc.mercier@mpg-partners.com*

22 03 2023

Introduction - Plan presentation

Motivations

- Introduce new generative methods based on kernel (RKHS theory).
- Illustrate our kernel library CODPY [1].
- Illustrate potential applications in Finance, with a **toy risk management** example.

[1] codpy user manual and installation guidelines at https://pypi.org/project/codpy/.

CODPY : Why developing a RKHS ML library ?

- Fast, comput. efficient methods, adapted to sparse input data. For instance, this presentation is a Jupyter notebook, fully customizable. The generation of this document should take **ten seconds** with a standard laptop.
- Quantifiable error methods (uncertainty quantification).
- $\implies$ efficient bridge to Optimal Transport tools (clustering / generative methods).
- Application range : Machine Learning, Statistic, numerical simulations (Mesh Free, Particle Methods).

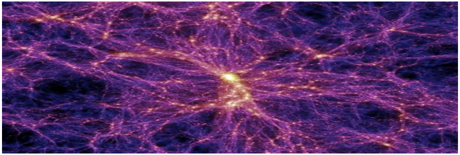


Figure 1: A particle method (SPH) simulation

Generative models with kernels : CelebA dataset

Generation of a variate, that is **statistically** coherent with historical observations.

Figure 2: Original (left) and generated (right) images of CelebA dataset. Image generation is **one second** on a standard laptop, decoding, learning time and matching included.

Generative kernels : Encoders and Decoders

Let $\mathbb{X}, \mathbb{Y}$ be **any** two probability measures, taking values into $\mathcal{X} \subset \mathbb{R}^{D_X}$,$\mathcal{Y} \subset \mathbb{R}^{D_Y}$.

Figure 3: Generating high res faces using low res cats

- $S : \mathcal{Y} \mapsto \mathcal{X}$ is the **encoder**. Example: faces res. $D_Y = 120000$.
- $\mathcal{X}$ is the **latent** space. example: cats res. $D_X = 100$.
- The inverse map $S^{-1} : \mathcal{X} \mapsto \mathcal{Y}$ is the **decoder**.
- The projection operator $z \mapsto (S^{-1} \circ S)(z)$ is called a **reconstruction**

How do we compute invertible maps ?

solving a generalized travelling salesman problem.

Generative kernels : Mathematical insights

Consider any map $S : \mathcal{Y} \mapsto \mathcal{X}$ transporting $\mathbb{Y}$ into $\mathbb{X}$. Optimal transport notations:

$$S_{\#}\mathbb{Y} = \mathbb{X} \Longleftrightarrow \int_{\mathbb{R}^{D_X}} \varphi(\cdot) d\mathbb{X} = \int_{\mathbb{R}^{D_Y}} \varphi \circ S(\cdot) d\mathbb{Y}, \forall \varphi \tag{1}$$

example : $\mathbb{X}$ uniform dist $\implies$ $\mathcal{Y}$ is described as a manifold of dim D_X (auto **features** extraction for **classification**).

Generalized travelling salesman problem: find S s.t.

$$\bar{S} = \arg \inf_{S_{\#}\mathbb{Y}=\mathbb{X}} \int_{\mathbb{R}^{D_X}} |\nabla S|^2 d\mathbb{X} \tag{2}$$

It is a problem of **diffusion type**. From a numerical point of view, it is a permutation finding problem, that is **NP-complete**. Thus we compute sub-optimal solutions.

Application settings - Macro presentation

- **Step1 - settings** : retrieve market data, set instruments, set pricers.
- **Step2 - generate market data** : consider a **agnostic / free** model to reproduce market data with generative methods. Simulate extreme market data for tomorrow date.
- **Step3 - learn the pricing functions:** use a predictive method to compute prices and greeks on generated data. $\mapsto$**real-time, basis-point accurate**, pricing and hedging solutions for large portfolios.
- **Step4 - analyse external risk factors** : use a generative methods to reverse the pricer function ($\sim$ reverse PnL). This will compute plausible market data scenario for any possible pricing values of our portfolio.

Application settings - Retrieve market data

Download real market data, retrieved from January 1, 2016 to December 31, 2021, for three assets: Google, Apple and Amazon.

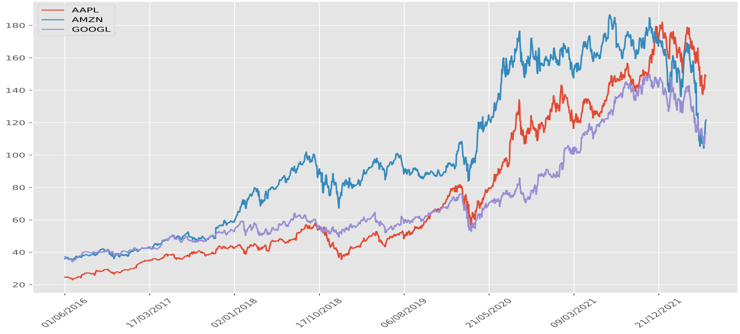
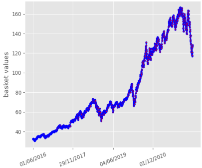


Figure 4: charts for Apple Amazon Google

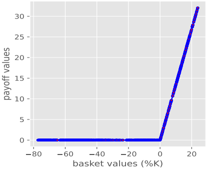
Application settings - portfolio settings

Set a portfolio : instruments (payoffs) and pricing engine (pricers)

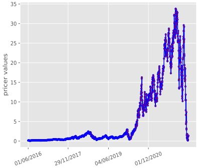
- Consider a payoff function $(t, x) \mapsto P(t, x) \in \mathbb{R}^{D_P}$, D_P number of instruments. Example, a **basket option** $P(t, x) = (< x, \omega > -K)^+$. $< x, \omega >$ **basket values**.
- Consider a pricing function $(t, x) \mapsto V(t, x) \in \mathbb{R}^{D_P}$, D_P number of instruments. Example, a **Black and Scholes formula** $V(t, x) = \mathbb{E}^{X_T}(P(T, \cdot)|X_t)$.



(a) Basket values as a function of time



(b) Payoff as a function of basket



(c) Pricer as a function of time

Synthetic data generation : fit a model to data

A model can be described by a stochastic differential equation (SDE). For instance the following **log-return** model is a solution to a SDE.:

$$X_t = X_s \exp((t-s)\mu + \sqrt{t-s}\mathbb{X}) \tag{3}$$

where $\mathbb{X}$ is an unknown distribution (Euler scheme).

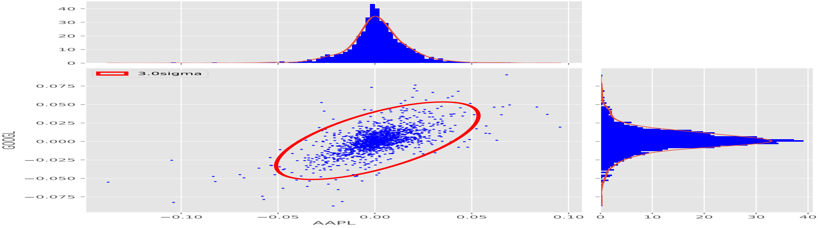


Figure 6: Log return distribution of historical market data

Synthetic data generation : generate samples

Example of a generated variate based upon the historical distribution $\mathbb{X}$

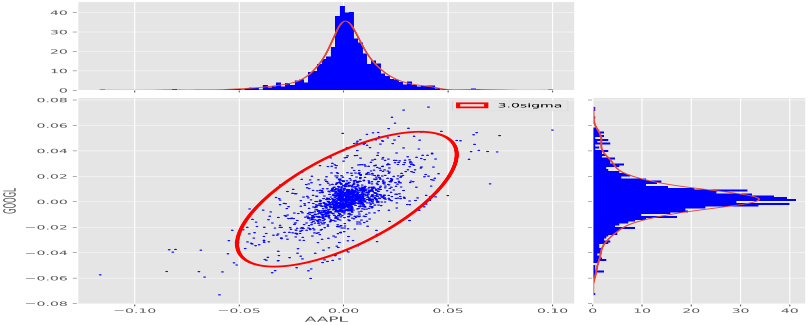


Figure 7: Log return distribution of generated market data

Synthetic generation : check generated samples

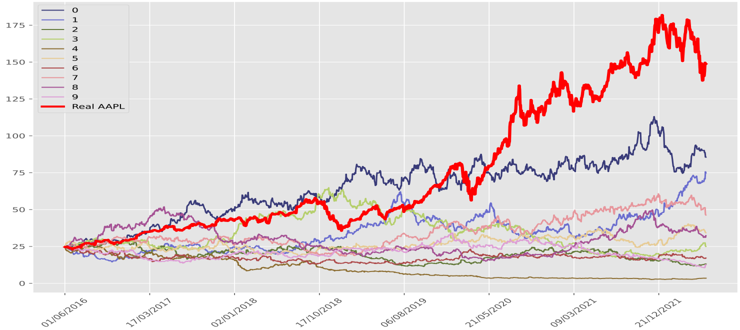
Computations of various statistical indicators, as the fourth moments and Kolmogorov-Smirnov tests.

Table 1: Stats for historical (generated) data

	AAPL	AMZN	GOOGL
Mean	0.0013(0.0015)	0.001(0.00078)	0.00073(0.00066)
Variance	-0.48(-0.23)	-0.14(-0.18)	-0.48(-0.36)
Skewness	0.0003(0.00031)	0.00031(0.00031)	0.00025(0.00028)
Kurtosis	7.4(4.1)	3.7(2.3)	6.5(3.3)
KS test	0.86(0.05)	0.7(0.05)	0.54(0.05)

Synthetic data generation : generate paths

Ten examples of re-generated paths. Applications: Back testing, Optimization of Monte-Carlo simulations, PDE pricers (for XVA), Asset Mgmt (portfolio allocation), ...



Predictive pricing methods : prediction errors

explanability / uncertainty quantification of kernel methods

- Step 1: learn the pricing functions on historical values (**blue** basket values).
- Step 2: predict the pricing functions on intraday values (**red** basket values simulated from the generative methods at date today plus 5 days).
- Step 3: compute errors (**iso-contours** = worst error bounds ($\sim$ confidence levels)).

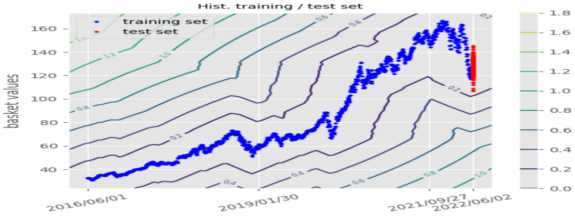


Figure 9: Error bounds on prices prediction.

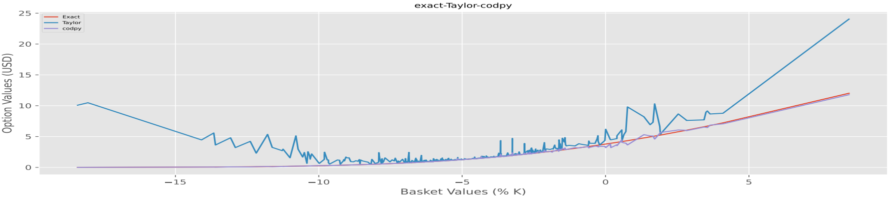
Predictive pricing methods

Real-time pricing solutions for large portfolios

Benchmark of two methods (real time pricing eligible methods)

- Taylor second order approximation.
- Kernel predictive machine.

exact = reference price (Black and Scholes).



Predictive pricing methods: predict greeks

Computed first order greeks

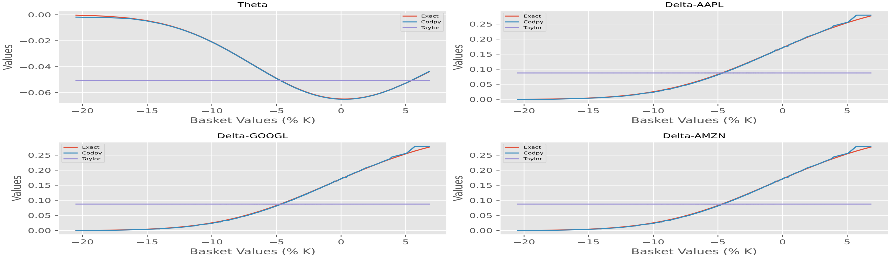
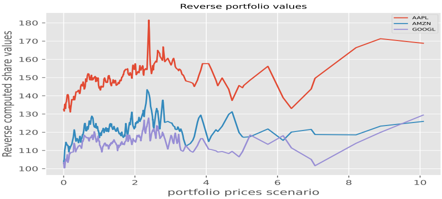


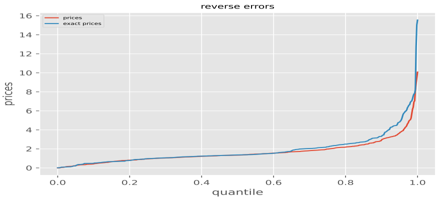
Figure 11: Greeks corrected

Variables explanation : exogenous risk factor analysis

Example : reverse prices (similar to **reverse PnL**)

- Encode our generated market data with our predicted prices.
- Generate new samples of the price distribution (blue - right figure).
- Decode these prices, producing new market data (left figure).
- Use our explicit pricer function to compute the exact price associated to these generated market data for benchmark (red - right figure).





Conclusions

What are the main points highlighted by this presentation

Kernel methods are not only efficient for numerical simulations, they are very competitive for machine learning and artificial intelligence purposes. For risk or portfolio management applications:

- Synthetic data generation : allow to model **any** given continuous random variables, eventually conditionally to other variables (e.g. analysis of customer's behaviors).
- Synthetic data generation allows to **revisit** existing quant. diffusion models, and propose a general method to **calibrate** them. More accurate risk sources modelling.
- Predictive methods can learn expensive computational methods as pricers from quite few discrete examples. **Fast real-time, basis point accurate** pricing and hedging solutions for large portfolios.
- Synthetic data generation allow to capture and analyse external risk factors impact reliabily.

Thank you