Outlining the Pricing of Cliquet-style Options: From Crude Monte Carlo Simulation to Statistical Machine Learning (Implemented in Python (v3.6.5))

Frank Oertel

Philosophy, Logic & Scientific Method
Centre for Philosophy of Natural and Social Sciences (CPNSS)
London School of Economics & Political Science, UK
`http://www.frank-oertel-math.de`

Department of Financial Mathematics
Fraunhofer-Institut für Techno- und Wirtschaftsmathematik

D-67663 Kaiserslautern

06.06.2018

Contents

1. Structure of Cliquet Options
2. Pricing of Cliquet Options by Crude Monte Carlo Simulation
3. Beyond Crude Monte Carlo: from Variance Reduction to Machine Learning

Payoff structure of Cliquet options I

Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $T > 0$ be a (non-random) future point in time and $n \in \mathbb{N}$. Subdivide the interval $[0, T)$ into n disjoint subintervals $\left[t_{k-1}, t_k\right)$, called reset periods of length $\Delta t_k := t_k - t_{k-1}$, where $t_n := T$ and $t_{k-1} := (k-1)\frac{T}{n}$ denotes the $(k-1)$'th reset day ($k = 1, \ldots, n$). Notice that each $\Delta t_k = \frac{T}{n}$ is equidistant.

Payoff structure of Cliquet options I

Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $T > 0$ be a (non-random) future point in time and $n \in \mathbb{N}$. Subdivide the interval $[0, T)$ into n disjoint subintervals $[t_{k-1}, t_k)$, called reset periods of length $\Delta t_k := t_k - t_{k-1}$, where $t_n := T$ and $t_{k-1} := (k-1)\frac{T}{n}$ denotes the $(k-1)$'th reset day ($k = 1, \ldots, n$). Notice that each $\Delta t_k = \frac{T}{n}$ is equidistant.

The relative return of an asset with stochastic price process $(S_t)_{0 \le t \le T}$ over a reset period $[t_{k-1}, t_k)$ is then defined (on Ω) as

$$R_k := \frac{S_{t_k} - S_{t_{k-1}}}{S_{t_{k-1}}} = \frac{S_{t_k}}{S_{t_{k-1}}} - 1 \,.$$

Payoff structure of Cliquet options I

Fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $T > 0$ be a (non-random) future point in time and $n \in \mathbb{N}$. Subdivide the interval $[0, T)$ into n disjoint subintervals $[t_{k-1}, t_k)$, called reset periods of length $\Delta t_k := t_k - t_{k-1}$, where $t_n := T$ and $t_{k-1} := (k-1)\frac{T}{n}$ denotes the $(k-1)$'th reset day ($k = 1, \ldots, n$). Notice that each $\Delta t_k = \frac{T}{n}$ is equidistant.

The relative return of an asset with stochastic price process $(S_t)_{0 \leq t \leq T}$ over a reset period $[t_{k-1}, t_k)$ is then defined (on Ω) as

$$R_k := \frac{S_{t_k} - S_{t_{k-1}}}{S_{t_{k-1}}} = \frac{S_{t_k}}{S_{t_{k-1}}} - 1 \,.$$

Let $c \geq 0$ and $g < nc$ (c is known as local cap and g as global floor) and $K > 0$. Let $R_k \wedge c \equiv \min\{R_k, c\}$ be the truncated relative return over a reset period $[t_{k-1}, t_k)$.

Payoff structure of Cliquet options II

Following Bernard and Li (cf. [1]) we consider the following two specific types of derivative payoffs (depending on n):

$$X_{n,T}(\psi) := K \max\left\{1+g, 1+\sum_{k=1}^{n} \psi(R_k, c)\right\}$$

where
$\mathbb{R} \times [0,\infty) \ni (x,c) \mapsto \psi(x,c) \in \{x \wedge c, (x \wedge c)^+\}$

Payoff structure of Cliquet options II

Following Bernard and Li (cf. [1]) we consider the following two specific types of derivative payoffs (depending on n):

$$X_{n,T}(\psi) := K \max\left\{1+g, 1+\sum_{k=1}^{n} \psi(R_k, c)\right\}$$

where
$\mathbb{R} \times [0, \infty) \ni (x, c) \mapsto \psi(x, c) \in \{x \wedge c, (x \wedge c)^+\} \stackrel{\checkmark}{=} \{x \wedge c, x^+ \wedge c\}$.

Payoff structure of Cliquet options II

Following Bernard and Li (cf. [1]) we consider the following two specific types of derivative payoffs (depending on n):

$$X_{n,T}(\psi) := K \max\left\{1+g, 1+\sum_{k=1}^{n} \psi(R_k, c)\right\}$$

where
$\mathbb{R} \times [0, \infty) \ni (x, c) \mapsto \psi(x, c) \in \{x \wedge c, \left(x \wedge c\right)^{+}\} \overset{\checkmark}{=} \{x \wedge c, x^{+} \wedge c\}$.
$\psi(R_k, c) := R_k \wedge c$ reflects the payoff of a *Monthly Sum Cap (MSC)*. If $\psi(R_k, c) := R_k^{+} \wedge c$ a *Minimum Coupon Cliquet (MCC)* is considered.

Payoff structure of Cliquet options II

Following Bernard and Li (cf. [1]) we consider the following two specific types of derivative payoffs (depending on n):

$$X_{n,T}(\psi) := K \max\left\{1 + g, 1 + \sum_{k=1}^{n} \psi(R_k, c)\right\}$$

where
$\mathbb{R} \times [0, \infty) \ni (x, c) \mapsto \psi(x, c) \in \{x \wedge c, (x \wedge c)^+\} \overset{\checkmark}{=} \{x \wedge c, x^+ \wedge c\}$.
$\psi(R_k, c) := R_k \wedge c$ reflects the payoff of a *Monthly Sum Cap (MSC)*. If $\psi(R_k, c) := R_k^+ \wedge c$ a *Minimum Coupon Cliquet (MCC)* is considered.

Notice that on Ω

$$1 + g \leq \frac{X_{n,T}(\psi)}{K} \leq \max\left\{1 + g, 1 + nc\right\} = 1 + n\,c\,.$$

Payoff structure of Cliquet options III

Obviously,

$$\frac{X_{n,T}}{K} = 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\}$$

Payoff structure of Cliquet options III

Obviously,

$$\frac{X_{n,T}}{K} = 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\}$$

$$=$$

Payoff structure of Cliquet options III

Obviously,

$$\frac{X_{n,T}}{K} = 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\}$$

$$= 1 + g + \max\left\{0, \sum_{k=1}^{n} \psi(R_k, c) - g\right\}$$

Payoff structure of Cliquet options III

Obviously,

$$
\begin{aligned}
\frac{X_{n,T}}{K} &= 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\} \\
&= 1 + g + \max\left\{0, \sum_{k=1}^{n} \psi(R_k, c) - g\right\} \\
&=
\end{aligned}
$$

Payoff structure of Cliquet options III

Obviously,

$$\begin{aligned}
\frac{X_{n,T}}{K} &= 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\} \\
&= 1 + g + \max\left\{0, \sum_{k=1}^{n} \psi(R_k, c) - g\right\} \\
&= 1 + g + \left(\sum_{k=1}^{n} Z_{n,k}(\psi)\right)^{+},
\end{aligned}$$

Payoff structure of Cliquet options III

Obviously,

$$\begin{aligned} \frac{X_{n,T}}{K} &= 1 + \max\left\{g, \sum_{k=1}^{n} \psi(R_k, c)\right\} \\ &= 1 + g + \max\left\{0, \sum_{k=1}^{n} \psi(R_k, c) - g\right\} \\ &= 1 + g + \left(\sum_{k=1}^{n} Z_{n,k}(\psi)\right)^{+}, \end{aligned}$$

where

$$Z_{n,k}(\psi) := \psi(R_k, c) - \frac{g}{n} \quad (k \in \{1, 2, \dots, n\}).$$

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

(S1) Simulate the stochastic dynamics of the underlying asset (i. e., simulate a possible “path”) by either using stochastic analytic methods (respectively Itô calculus) directly (to solve related SDEs analytically if possible) or making use of the Euler scheme.

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

(S1) Simulate the stochastic dynamics of the underlying asset (i. e., simulate a possible “path”) by either using stochastic analytic methods (respectively Itô calculus) directly (to solve related SDEs analytically if possible) or making use of the Euler scheme.

(S2) Calculate the payoff of the derivative security on each path.

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

(S1) Simulate the stochastic dynamics of the underlying asset (i. e., simulate a possible "path") by either using stochastic analytic methods (respectively Itô calculus) directly (to solve related SDEs analytically if possible) or making use of the Euler scheme.

(S2) Calculate the payoff of the derivative security on each path.

(S3) Discount the payoff at the risk-free rate.

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

(S1) Simulate the stochastic dynamics of the underlying asset (i. e., simulate a possible “path”) by either using stochastic analytic methods (respectively Itô calculus) directly (to solve related SDEs analytically if possible) or making use of the Euler scheme.

(S2) Calculate the payoff of the derivative security on each path.

(S3) Discount the payoff at the risk-free rate.

(S4) Calculate the average over all simulated paths.

Pricing of Cliquet options by crude Monte Carlo simulation I

In general, crude Monte Carlo simulation applied to *path-dependent* derivative securities works as follows:

(S1) Simulate the stochastic dynamics of the underlying asset (i. e., simulate a possible "path") by either using stochastic analytic methods (respectively Itô calculus) directly (to solve related SDEs analytically if possible) or making use of the Euler scheme.

(S2) Calculate the payoff of the derivative security on each path.

(S3) Discount the payoff at the risk-free rate.

(S4) Calculate the average over all simulated paths.

(S5) Apply error reduction methods.

Pricing of Cliquet options by crude Monte Carlo simulation II

The following result shows us that we may apply the crude Monte-Carlo method to the pricing of such Cliquet-style options:

Pricing of Cliquet options by crude Monte Carlo simulation II

The following result shows us that we may apply the crude Monte-Carlo method to the pricing of such Cliquet-style options:

Lemma

Assume that we are working in a complete standard Black-Merton-Scholes financial market model without arbitrage and unique martingale measure $\mathbb{Q}$. Let $r > 0$ be the "risk-free" interest rate, $\eta \geq 0$ the yield of dividend and $\sigma > 0$ the constant "volatility". Let $n \in \mathbb{N}$. Then the random variables $Z_{n,1}(\psi), Z_{n,2}(\psi), \ldots, Z_{n,n}(\psi)$ are i.i.d., and

$$Z_{n,k}(\psi) \stackrel{d}{=} \tilde{Z}_n(\psi) := \psi\left(\exp\left(\sigma\sqrt{\frac{T}{n}}X + \left(r - \eta - \frac{1}{2}\sigma^2\right)\frac{T}{n} - 1\right), c\right) - \frac{g}{n},$$

where $X \sim N(0,1)$.

Pricing of Cliquet options by crude Monte Carlo simulation II

Proof.

Fix $t \geq 0$ and $n \in \mathbb{N}$. Let $S_0 > 0$ be the (observed) initial (non-random) price of the asset. Given our financial market model assumption it follows that

$$\begin{aligned} S_t &= S_0\, \mathcal{E}\big(R\big)_t = S_0 \exp\Big(R_t - \frac{1}{2}[R,R]_t\Big) \\ &= S_0 \exp\Big(\sigma W_t + \big(r - \eta - \frac{1}{2}\sigma^2\big)\, t\Big), \end{aligned}$$

where $(W_t)_{t\geq 0}$ is a standard Brownian motion (under $\mathbb{Q}$) and $R_t := (r-\eta)t + \sigma W_t$ denotes the time t-value of the stochastic dividend adjusted relative return process.

Pricing of Cliquet options by crude Monte Carlo simulation II

Proof ctd.

Thus,

$$R_k + 1 = \exp\Big(\sigma\big(W_{t_k} - W_{t_{k-1}}\big) + \big(r - \eta - \frac{1}{2}\sigma^2\big)\big(t_k - t_{k-1}\big)\Big)$$

for all $k \in \{1, 2, \ldots, n\}$, implying that the random variables $R_1, R_2, \ldots, R_n$ and *hence* the random variables $Z_{n,1}(\psi), Z_{n,2}(\psi), \ldots, Z_{n,n}(\psi)$ are independent (since g is deterministic).

Pricing of Cliquet options by crude Monte Carlo simulation II

Proof ctd.
Thus,

$$R_k + 1 = \exp\left(\sigma\left(W_{t_k} - W_{t_{k-1}}\right) + \left(r - \eta - \frac{1}{2}\sigma^2\right)\left(t_k - t_{k-1}\right)\right)$$

for all $k \in \{1, 2, \ldots, n\}$, implying that the random variables $R_1, R_2, \ldots, R_n$ and *hence* the random variables $Z_{n,1}(\psi), Z_{n,2}(\psi), \ldots, Z_{n,n}(\psi)$ are independent (since g is deterministic). Moreover,

$$W_{t_k} - W_{t_{k-1}} \stackrel{d}{=} W_{t_k - t_{k-1}} = W_{\frac{T}{n}} \stackrel{d}{=} \sqrt{\frac{T}{n}} W_1 \, ,$$

and the claim follows. □

Pricing of Cliquet options by crude Monte Carlo simulation III

To approximate the "risk-neutral" time-0 price

$$p_0(X_{n,T}) := \exp(-rT)\mathbb{E}_{\mathbb{Q}}\left[X_{n,T}\right]$$

by a crude Monte Carlo simulation method we have to perform the following steps:

Pricing of Cliquet options by crude Monte Carlo simulation III

(i) Fix a “large” number $m \in \mathbb{N}$ (of paths of the underlying); $m = 100000$, say.

Pricing of Cliquet options by crude Monte Carlo simulation III

(i) Fix a "large" number $m \in \mathbb{N}$ (of paths of the underlying); $m = 100000$, say.

(ii) Fix $\nu \in \{1, 2, \ldots, m\}$. Generate a random sample of n i.i.d standard-normally distributed random variables $X_1^{(\nu)}, X_2^{(\nu)}, \ldots, X_n^{(\nu)}$ and consider the simulated i.i.d random variables $\tilde{Z}_{n,1}^{(\nu)}(\psi), \tilde{Z}_{n,2}^{(\nu)}(\psi), \ldots, \tilde{Z}_{n,n}^{(\nu)}(\psi)$, where

$$\tilde{Z}_{n,k}^{(\nu)}(\psi) := \psi\Big(\exp\Big(\sigma\sqrt{\frac{T}{n}}X_k^{(\nu)} + \big(r - \eta - \frac{1}{2}\sigma^2\big)\frac{T}{n}\Big) - 1, c\Big) - \frac{g}{n}$$

Pricing of Cliquet options by crude Monte Carlo simulation III

(i) Fix a "large" number $m \in \mathbb{N}$ (of paths of the underlying); $m = 100000$, say.

(ii) Fix $\nu \in \{1, 2, \ldots, m\}$. Generate a random sample of n i.i.d standard-normally distributed random variables $X_1^{(\nu)}, X_2^{(\nu)}, \ldots, X_n^{(\nu)}$ and consider the simulated i.i.d random variables $\tilde{Z}_{n,1}^{(\nu)}(\psi), \tilde{Z}_{n,2}^{(\nu)}(\psi), \ldots, \tilde{Z}_{n,n}^{(\nu)}(\psi)$, where

$$\tilde{Z}_{n,k}^{(\nu)}(\psi) := \psi\Big(\exp\Big(\sigma\sqrt{\frac{T}{n}} X_k^{(\nu)} + \big(r - \eta - \frac{1}{2}\sigma^2\big)\frac{T}{n}\Big) - 1, c\Big) - \frac{g}{n}$$

Calculate the simulated payoff

$$X_{n,T}^{(\nu)}(\psi) := K\Big(1 + g + \Big(\sum_{k=1}^{n} \tilde{Z}_{n,k}^{(\nu)}(\psi)\Big)^+\Big)$$

(iii) Repeat step (ii) m-times independently.

Pricing of Cliquet options by crude Monte Carlo simulation III

(i) Fix a "large" number $m \in \mathbb{N}$ (of paths of the underlying); $m = 100000$, say.

(ii) Fix $\nu \in \{1, 2, \ldots, m\}$. Generate a random sample of n i.i.d standard-normally distributed random variables $X_1^{(\nu)}, X_2^{(\nu)}, \ldots, X_n^{(\nu)}$ and consider the simulated i.i.d random variables $\tilde{Z}_{n,1}^{(\nu)}(\psi), \tilde{Z}_{n,2}^{(\nu)}(\psi), \ldots, \tilde{Z}_{n,n}^{(\nu)}(\psi)$, where

$$\tilde{Z}_{n,k}^{(\nu)}(\psi) := \psi\Big(\exp\Big(\sigma\sqrt{\frac{T}{n}}X_k^{(\nu)} + \big(r - \eta - \frac{1}{2}\sigma^2\big)\frac{T}{n}\Big) - 1, c\Big) - \frac{g}{n}$$

Calculate the simulated payoff

$$X_{n,T}^{(\nu)}(\psi) := K\Big(1 + g + \Big(\sum_{k=1}^{n} \tilde{Z}_{n,k}^{(\nu)}(\psi)\Big)^+\Big)$$

(iii) Repeat step (ii) m-times independently.

(iv) Calculate $\frac{1}{m}\sum_{\nu=1}^{m} X_{n,T}^{(\nu)}(\psi)$.

Pricing of Cliquet options by crude Monte Carlo simulation III

Observation

Steps (ii) and (iii) imply that in fact all $m \cdot n$ random variables $\tilde{Z}^{(1)}_{n,1}(\psi), \ldots, \tilde{Z}^{(1)}_{n,n}(\psi), \tilde{Z}^{(2)}_{n,1}(\psi), \ldots, \tilde{Z}^{(2)}_{n,n}(\psi), \ldots, \tilde{Z}^{(m)}_{n,1}(\psi), \ldots, \tilde{Z}^{(m)}_{n,n}(\psi)$ are i.i.d standard-normally distributed random variables. In other words,

$$\left(\tilde{Z}^{(1)}_{n,1}(\psi), \ldots, \tilde{Z}^{(1)}_{n,n}(\psi), \tilde{Z}^{(2)}_{n,1}(\psi), \ldots, \tilde{Z}^{(2)}_{n,n}(\psi), \ldots, \tilde{Z}^{(m)}_{n,1}(\psi), \ldots, \tilde{Z}^{(m)}_{n,n}(\psi)\right)^{\top}$$

is an $m \cdot n$-dimensional standard Gaussian random vector.

Pricing of Cliquet options by crude Monte Carlo simulation III

Observation

Steps (ii) and (iii) imply that in fact all $m \cdot n$ random variables $\tilde{Z}^{(1)}_{n,1}(\psi), \ldots, \tilde{Z}^{(1)}_{n,n}(\psi), \tilde{Z}^{(2)}_{n,1}(\psi), \ldots, \tilde{Z}^{(2)}_{n,n}(\psi), \ldots, \tilde{Z}^{(m)}_{n,1}(\psi), \ldots, \tilde{Z}^{(m)}_{n,n}(\psi)$ are i.i.d standard-normally distributed random variables. In other words,

$$\left(\tilde{Z}^{(1)}_{n,1}(\psi), \ldots, \tilde{Z}^{(1)}_{n,n}(\psi), \tilde{Z}^{(2)}_{n,1}(\psi), \ldots, \tilde{Z}^{(2)}_{n,n}(\psi), \ldots, \tilde{Z}^{(m)}_{n,1}(\psi), \ldots, \tilde{Z}^{(m)}_{n,n}(\psi)\right)^\top$$

is an $m \cdot n$-dimensional standard Gaussian random vector.

Morover, we have:

Pricing of Cliquet options by crude Monte Carlo simulation III

Observation

Fix $\nu \in \{1, 2, \ldots m\}$*. Since*

$$\sum_{k=1}^{n} \tilde{Z}_{n,k}^{(\nu)}(\psi) = \sqrt{n}\Big(\frac{1}{\sqrt{n}} \sum_{k=1}^{n} \tilde{Z}_{n,k}^{(\nu)}(\psi)\Big) \stackrel{d}{=} \sqrt{n}\, Z$$

for some $Z \sim N(0,1)$ *we can apply the Strong Law of Large Numbers (SLLN), implying that in fact*

$$\frac{1}{m} \sum_{\nu=1}^{m} X_{n,T}^{(\nu)}(\psi) \stackrel{m\to\infty}{\longrightarrow} \mathbb{E}_{\mathbb{Q}}\big[X_{n,T}^{(1)}(\psi)\big] \quad \mathbb{Q}-\textit{a.s.}$$

How the efficiency of Monte Carlo simulation can be improved I

The crude Monte-Carlo method is a useful approach to estimate numerically the value of integrals over domains of integration in $\mathbb{R}^n$ for large values of n.

How the efficiency of Monte Carlo simulation can be improved I

The crude Monte-Carlo method is a useful approach to estimate numerically the value of integrals over domains of integration in $\mathbb{R}^n$ for large values of n.

A direct implication of the Central Limit Theorem (CLT) and the Strong Law of Large Numbers (SLLN) shows that the resulting error asymptotically converges at rate $O(\frac{1}{\sqrt{n}})$ - independent of the dimension n of the underlying domain of integration.

How the efficiency of Monte Carlo simulation can be improved I

The crude Monte-Carlo method is a useful approach to estimate numerically the value of integrals over domains of integration in $\mathbb{R}^n$ for large values of n.

A direct implication of the Central Limit Theorem (CLT) and the Strong Law of Large Numbers (SLLN) shows that the resulting error asymptotically converges at rate $O(\frac{1}{\sqrt{n}})$ - independent of the dimension n of the underlying domain of integration.

Moreover, the convergence rate in the Monte-Carlo method is strongly influenced by a variance term which depends on the integrand and the probability density function of the sampling distribution that is used.

How the efficiency of Monte Carlo simulation can be improved I

The crude Monte-Carlo method is a useful approach to estimate numerically the value of integrals over domains of integration in $\mathbb{R}^n$ for large values of n.

A direct implication of the Central Limit Theorem (CLT) and the Strong Law of Large Numbers (SLLN) shows that the resulting error asymptotically converges at rate $O(\frac{1}{\sqrt{n}})$ - independent of the dimension n of the underlying domain of integration. Moreover, the convergence rate in the Monte-Carlo method is strongly influenced by a variance term which depends on the integrand and the probability density function of the sampling distribution that is used. So, we are interested in reducing the impact of this variance term and look for suitable variance reduction techniques.

How the efficiency of Monte Carlo simulation can be improved II

How the efficiency of Monte Carlo simulation can be improved II

- **Variance reducing correlations.** One approach to reduce the variance of the Monte-Carlo estimate is to develop a corresponding estimate based on a sequence of non-i.i.d. random variables $(X_i)_{1 \le i \le n}$ which are correlated in such a way that these correlations lead to cancellations in the (approximating) sum, yielding a smaller variance term for the estimate.

How the efficiency of Monte Carlo simulation can be improved II

- **Variance reducing correlations.** One approach to reduce the variance of the Monte-Carlo estimate is to develop a corresponding estimate based on a sequence of non-i.i.d. random variables $(X_i)_{1 \le i \le n}$ which are correlated in such a way that these correlations lead to cancellations in the (approximating) sum, yielding a smaller variance term for the estimate. To justify this approach we have to work with a version of the SLLN that allows us to include the case of sequences of correlated random variables (key word: *correlation bounds*). Moreover, quantile transformation techniques can be applied to generate such correlations.

How the efficiency of Monte Carlo simulation can be improved III

- **Control variate technique.** We want to compute $\mathbb{E}_{\mathbb{Q}}[X]$ for some $X \in L^1(\mathbb{Q})$. Suppose that an explicit analytic expression for $\mathbb{E}_{\mathbb{Q}}[X]$ is not available (yet).

How the efficiency of Monte Carlo simulation can be improved III

- **Control variate technique.** We want to compute $\mathbb{E}_{\mathbb{Q}}[X]$ for some $X \in L^1(\mathbb{Q})$. Suppose that an explicit analytic expression for $\mathbb{E}_{\mathbb{Q}}[X]$ is not available (yet). Further assume that there exists some $X^* \in L^1(\mathbb{Q})$ such that $\alpha^* := \mathbb{E}_{\mathbb{Q}}[X^*]$ can be integrated analytically.

How the efficiency of Monte Carlo simulation can be improved III

- **Control variate technique.** We want to compute $\mathbb{E}_{\mathbb{Q}}[X]$ for some $X \in L^1(\mathbb{Q})$. Suppose that an explicit analytic expression for $\mathbb{E}_{\mathbb{Q}}[X]$ is not available (yet). Further assume that there exists some $X^* \in L^1(\mathbb{Q})$ such that $\alpha^* := \mathbb{E}_{\mathbb{Q}}[X^*]$ can be integrated analytically. If the variance error term in the Monte Carlo estimate of $\mathbb{E}_{\mathbb{Q}}[X - X^*]$ were smaller than that one of $\mathbb{E}_{\mathbb{Q}}[X]$ the variance error term of $\mathbb{E}_{\mathbb{Q}}[X] = \mathbb{E}_{\mathbb{Q}}[X - X^*] + \alpha^*$ would be larger than the sum of α^* and the variance error term of the Monte Carlo estimate of $\mathbb{E}_{\mathbb{Q}}[X - X^*]$.

How the efficiency of Monte Carlo simulation can be improved IV

How the efficiency of Monte Carlo simulation can be improved IV

- **Importance sampling.** Choose only “important” paths; i. e., trajectories so that regions which contribute significantly to $\mathbb{E}_{\mathbb{Q}}[X]$ are sampled with larger frequency ($\rightsquigarrow$ Look for region parts where the value of the payoff is non-negative and perform simulations over these parts).

How the efficiency of Monte Carlo simulation can be improved IV

- **Importance sampling.** Choose only "important" paths; i. e., trajectories so that regions which contribute significantly to $\mathbb{E}_{\mathbb{Q}}[X]$ are sampled with larger frequency ($\rightsquigarrow$ Look for region parts where the value of the payoff is non-negative and perform simulations over these parts).
- **Conditional expectation trick.** Also by making use of Jensen's inequality and standard properties of *conditional* expectation operators one can reduce the variance of the Monte-Carlo estimate.

How the efficiency of Monte Carlo simulation can be improved V

Further non-trivial generalisations of crude Monte Carlo simulation are:

How the efficiency of Monte Carlo simulation can be improved V

Further non-trivial generalisations of crude Monte Carlo simulation are:

- **Nested Monte Carlo (NMC).** NMC is very important (if not even required !) for simulating the stochastic IMM exposure in counterparty risk as one ingredient of the advanced CVA capital charge in Basel III (cf. [7]).

How the efficiency of Monte Carlo simulation can be improved V

Further non-trivial generalisations of crude Monte Carlo simulation are:

- **Nested Monte Carlo (NMC).** NMC is very important (if not even required !) for simulating the stochastic IMM exposure in counterparty risk as one ingredient of the advanced CVA capital charge in Basel III (cf. [7]). However, NMC requires complex computations at portfolio level (netting sets) and requires GPU parallel computing.

How the efficiency of Monte Carlo simulation can be improved V

Further non-trivial generalisations of crude Monte Carlo simulation are:

- **Nested Monte Carlo (NMC).** NMC is very important (if not even required !) for simulating the stochastic IMM exposure in counterparty risk as one ingredient of the advanced CVA capital charge in Basel III (cf. [7]). However, NMC requires complex computations at portfolio level (netting sets) and requires GPU parallel computing.
- **Machine learning of NMC.** This powerful machinery can be used to manoeuvre some nonlinear factors that may be unfolded by stochastic approximation algorithms (such as Longstaff-Schwartz) and to solve high-dimensional non-convex XVA minimisation tasks.

How the efficiency of Monte Carlo simulation can be improved V

Further non-trivial generalisations of crude Monte Carlo simulation are:

- **Nested Monte Carlo (NMC).** NMC is very important (if not even required !) for simulating the stochastic IMM exposure in counterparty risk as one ingredient of the advanced CVA capital charge in Basel III (cf. [7]). However, NMC requires complex computations at portfolio level (netting sets) and requires GPU parallel computing.
- **Machine learning of NMC.** This powerful machinery can be used to manoeuvre some nonlinear factors that may be unfolded by stochastic approximation algorithms (such as Longstaff-Schwartz) and to solve high-dimensional non-convex XVA minimisation tasks.

Let us shed a slightly bit of light on the latter subject.

Post financial crisis challenges 2018 and beyond - and an emergence

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.
- Rough stochastic volatility models: when fractional Brownian motion unfolds in market data.

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.
- Rough stochastic volatility models: when fractional Brownian motion unfolds in market data.

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.
- Rough stochastic volatility models: when fractional Brownian motion unfolds in market data.

Are these fascinating, late-breaking and mathematically challenging problems contained in something “larger”?

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.
- Rough stochastic volatility models: when fractional Brownian motion unfolds in market data.

Are these fascinating, late-breaking and mathematically challenging problems contained in something "larger"? Very likely, this is the case!

Post financial crisis challenges 2018 and beyond - and an emergence

- Central clearing and modelling of systemic risk: a network approach.
- Quantification of model risk: SR 11-7 and its impacts.
- Rough stochastic volatility models: when fractional Brownian motion unfolds in market data.

Are these fascinating, late-breaking and mathematically challenging problems contained in something “larger”? Very likely, this is the case!

- Statistical machine learning, applied to complex derivatives pricing, hedging and risk measurement in the realm of model and counterparty credit risk, collateralisation and funding.

Statistical Machine Learning I

- Comparing statistics and machine learning very roughly, the former is strongly related to an analysis of finite samples, to model misspecification and to computational (distribution driven) tasks (such as maximum likelihood estimation (MLE)) while the latter is used for a distribution free modelling, prediction and steering of *large* uncertain data driven tasks, induced by *training* examples as input of “learning” artificial neural networks.

Statistical Machine Learning I

- Comparing statistics and machine learning very roughly, the former is strongly related to an analysis of finite samples, to model misspecification and to computational (distribution driven) tasks (such as maximum likelihood estimation (MLE)) while the latter is used for a distribution free modelling, prediction and steering of *large* uncertain data driven tasks, induced by *training* examples as input of “learning” artificial neural networks.
- A proper understanding of the underlying mathematics of machine learning particularly requires a strong inclusion and interplay of abstract analysis, convex and combinatorial optimisation, probability theory and complexity theory

Statistical Machine Learning I

- Comparing statistics and machine learning very roughly, the former is strongly related to an analysis of finite samples, to model misspecification and to computational (distribution driven) tasks (such as maximum likelihood estimation (MLE)) while the latter is used for a distribution free modelling, prediction and steering of *large* uncertain data driven tasks, induced by *training* examples as input of "learning" artificial neural networks.
- A proper understanding of the underlying mathematics of machine learning particularly requires a strong inclusion and interplay of abstract analysis, convex and combinatorial optimisation, probability theory and complexity theory ($\rightsquigarrow$ "curse of dimensionality").

Statistical Machine Learning I

- Comparing statistics and machine learning very roughly, the former is strongly related to an analysis of finite samples, to model misspecification and to computational (distribution driven) tasks (such as maximum likelihood estimation (MLE)) while the latter is used for a distribution free modelling, prediction and steering of *large* uncertain data driven tasks, induced by *training* examples as input of "learning" artificial neural networks.
- A proper understanding of the underlying mathematics of machine learning particularly requires a strong inclusion and interplay of abstract analysis, convex and combinatorial optimisation, probability theory and complexity theory ($\rightsquigarrow$ "curse of dimensionality").
- Statistical machine learning (SML), dealing primarily with the complexity of high-dimensional data, lies at the interface of statistics and machine learning.

Statistical Machine Learning II

A few SML key words:

Statistical Machine Learning II

A few SML key words:

- **Deep learning.** A set of learning algorithms designed to train multilayered neural networks (used by Amazon, Google (*TensorFlow*), Microsoft and Facebook and first global investment banks).

Statistical Machine Learning II

A few SML key words:

- **Deep learning.** A set of learning algorithms designed to train multilayered neural networks (used by Amazon, Google (*TensorFlow*), Microsoft and Facebook and first global investment banks).
- **Supervised learning.** Construction of a predictive model from a random variable X to an output variable Y, based on a finite set of i.i.d. observations $(X_i, Y_i)_{1 \leq i \leq n}$ ("predict output from input"), built on the sum of a data fitting "cost function" term and a norm regularisation term, where the used norm is l_1^n, respectively l_2^n for large n

Statistical Machine Learning II

A few SML key words:

- **Deep learning.** A set of learning algorithms designed to train multilayered neural networks (used by Amazon, Google (*TensorFlow*), Microsoft and Facebook and first global investment banks).
- **Supervised learning.** Construction of a predictive model from a random variable X to an output variable Y, based on a finite set of i.i.d. observations $(X_i, Y_i)_{1 \le i \le n}$ ("predict output from input"), built on the sum of a data fitting "cost function" term and a norm regularisation term, where the used norm is l_1^n, respectively l_2^n for large $n \rightsquigarrow$ convex and combinatorial optimisation, approximation theory, kernel methods - hence computational functional analysis.

Statistical Machine Learning III

Statistical Machine Learning III

- **Unsupervised learning.** Finding pattern in large unstructured data sets $\rightsquigarrow$ also mathematically challenging! *Includes* methods from algebraic geometry, spectral theory (high-dimensional SVD !), graph theory, matrix analysis (sparsity, matrix data completion, random matrices), Fourier analysis on the Boolean hypercube and cone programming...

Statistical Machine Learning III

- **Unsupervised learning.** Finding pattern in large unstructured data sets $\rightsquigarrow$ also mathematically challenging! *Includes* methods from algebraic geometry, spectral theory (high-dimensional SVD !), graph theory, matrix analysis (sparsity, matrix data completion, random matrices), Fourier analysis on the Boolean hypercube and cone programming...

Most recent applications of SML in the financial industry:

Statistical Machine Learning IV

- **Deep pricing:** use of neural networks and supervised learning to deliver efficient regression algorithms, such as the Longstaff-Schwartz algorithm, used for the pricing of (embedded) American options.

Statistical Machine Learning IV

- **Deep pricing:** use of neural networks and supervised learning to deliver efficient regression algorithms, such as the Longstaff-Schwartz algorithm, used for the pricing of (embedded) American options.
- **Deep hedging:** Use of multilayer perceptrons (MLPs) to approximate complex hedging strategies, subject to boundary conditions, such as transaction costs, liquidity constraints, collateralisation, margining and funding (XVAs !), legally binding regulatory demands (Basel III, Basel IV, SR 11-7 (Fed), CRR (EU)) and statutory audit requirements (such as IFRS 9 and IFRS 13).

Statistical Machine Learning IV

- **Deep pricing:** use of neural networks and supervised learning to deliver efficient regression algorithms, such as the Longstaff-Schwartz algorithm, used for the pricing of (embedded) American options.
- **Deep hedging:** Use of multilayer perceptrons (MLPs) to approximate complex hedging strategies, subject to boundary conditions, such as transaction costs, liquidity constraints, collateralisation, margining and funding (XVAs !), legally binding regulatory demands (Basel III, Basel IV, SR 11-7 (Fed), CRR (EU)) and statutory audit requirements (such as IFRS 9 and IFRS 13).
- **Deep calibration:** Use of SML techniques to approximate the solution of high-dimensional *inverse* problems, originating from a calibration of stochastic derivatives pricing models to given *and available* market data (e. g., stochastic volatility models).

A few references I

[1] G. Batres-Estrada
Deep Learning for Multivariate Financial Time Series.
MSc thesis, KTH Royal Institute of Technology (2015).

[2] C. Bernard, and W. V. Li.
Pricing and hedging of cliquet options and locally-capped contracts.
SIAM J. Fin Math **4**, *pp. 353-371* (2013).

[3] M. Broadie, Y. Du, and C. C. Moallemi.
Risk estimation via regression.
Oper. Res., **63***(5), pp. 1077-1097* (2015).

A few references II

[4] Yu-fen Chiu, and Ming-hua Hsieh.
Monte Carlo methods for valuation of ratchet equity indexed annuities.
Proceedings of the 2007 Winter Simulation Conference, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds (2008).

[5] R. Culkin, and S. R. Das.
Machine Learning in Finance: The Case of Deep Learning for Option Pricing.
Preprint, Santa Clara University (2017).

[6] J. M. Hutchinson, A. W. Lo, and T. Poggio.
A nonparametric approach to pricing and hedging derivative securities via learning networks.
J. of Finance **49** *(3), 851-889 (1994)*.

A few references III

[7] M. Kjaer.
On the Pricing of Cliquet Options with Global Floor and Cap.
Thesis for the Degree of Licentiate of Engineering, Chalmers University of Technology and Göteborg University (2004).

[8] R. Korn, E. Korn, and G. Kroisandt.
Monte Carlo methods and models in finance and insurance.
Chapman & Hall/CRC Financial Mathematics Series, Boca Raton (2010).

[9] R. Korn, B. Z. Temocin, and J. Wenzel.
Applications of the central limit theorem for pricing Cliquet-style options.
Eur. Actuar. J. **7**(2), pp. 465-480 (2017).

A few references IV

[10] Q. Liu.
Calculation of Credit Valuation Adjustment Based on Least Square Monte Carlo Methods.
Hindawi Publishing Corporation, Mathematical Problems in Engineering, Article ID 959312 (2015).

[11] P. Warken.
Effective Pricing of Cliquet Options.
Master's Thesis, University of Kaiserslautern (2015).

Thank you!

Thank you!

Questions? Remarks?